Navigation and Metaphors Section

Navigation and Metaphors

Overview

Key elements of a user interface are organization, navigation, metaphors, and visual style. Together these elements help the user form an overall conceptual model of the system, where information is located, what functions are available, and how to use the system.

Organization

Organization of information or data is concerned with how information is divided into topics, the sequence of the topics, and the interrelationships among the tropics. It also is concerned with the information and steps necessary to accomplish a task. Terminology, including the names of topics, subtopics, and tasks, is also very important.

The nature of the information, or data, and the tasks that manipulate the data will determine the organization of the data and tasks. For example, people use a telephone book to find the telephone number of a person. If the reader searches through the data looking for the phone number of a particular person, then the best way to organize the data is alphabetical by the last name and then the first name. As an additional example, an instruction book accompanying a bicycle that must be assembled will be organized in the sequence that best accomplishes this task, clearly indicated by step numbers.

Navigational Devices

The nature of the data and its inherent organization will determine those navigational devices that are the most appropriate and intuitive for the user. Navigational devices help the user know where they are, where they can go, and how they can get there. Navigational devices inappropriate to the organization of the data will confuse the user.

Books have inherent navigational devices. The visual appearance of the book provides affordances, or clues as to how to use the book. The user can very quickly see where the book begins and ends, assume how much information is in the book, determine how difficult or interesting the information is, all by merely glancing at the book without really reading it. They expect a table of contents to be at the beginning of the book,

providing a quick overview of the contents of the book. They expect an index to be at the back of the book, so they can easily search for specific data. They expect the book to have page numbers, and for those page numbers to be located on every page, either at the top or bottom of the page. They expect the information in the book to be broken up into chapters, and perhaps appendices. The chapter names may also be at the top or bottom of the pages. Books may have additional navigational devices, such as tabs at the side of the book, allowing the user to quickly see the chapters and go to a particular chapter. Sometimes people add their own navigational devices to books, such as pieces of paper placed between pages as bookmarks.

Metaphors

The user interface creates a virtual environment for the user, forming a visual and functional context for their actions, making the organizational structure of the computer system visible and accessible. Using metaphors (real-world objects, actions and terminology) helps the user transfer knowledge from the real-world to the virtual world. Well-chosen metaphors provide familiar objects and actions that allow users to predict the results of their actions in advance.

Metaphors should be simple and not require the user to learn and remember many rules and procedures. Some metaphors may not be appropriate for the task, not providing any advantages to the user, and in fact, create unnecessary confusion. An example of an inappropriate metaphor is requiring the user to place a diskette in the trash can to remove a diskette in the Macintosh OS. An example of an appropriate metaphor is the alphabet as it applies to the phone book example above.

Visual Style

Every user interface has a visual style, whether the designer intended for it to or not. Visual style is the combination of elements of the screen that establishes the look and feel of the user interface. Some of these elements include color, white space, contrast, scale, visual hierarchy, visual language, typography, and layout. Combined, these elements speak loud and clear, establishing the user's first impression of the Windows or web application, and contributing to (or distracting from) the ease of use of the application. An effective visual style supports the information architecture, navigation and metaphors.

A Windows interface has a very distinctive visual style: title bar at the top, menus underneath with menu items in a prescribed, consistent sequence, tool bars under the menu bar with text and icons, the document under the tool, status bar at the bottom, and so on. Web browsers have a distinctive visual style: menus at the top, a text entry box for the URL, the browser logo at the right also serving as an indicator that data is being downloaded, security and connection icons at the bottom left, and so on. On the Web, the visual style of a well-designed web site subtly assures the user that they are still on the site, and often where they are on the site.

General Navigation and Metaphor Standards

The following navigation and metaphor standards are to be adopted across all CDC Windows and web based surveillance applications. These standards will help ensure that

all CDC surveillance applications will have a consistent look and feel, intuitive navigation, and utilize common metaphors.

Provide keyboard access to all basic operations.

All functions with the exception of graphical functions should be accessible with the keyboard alone.

Provide mouse access to all basic operations.

All functions with the exception of text entry should be accessible with the mouse alone.

Use right mouse button for popup menus only.

Always make sure that the right mouse button is used for popup menus, also know as context menus, and not for anything else. Popup menus provide an efficient way for the user to access the commands that are associated with a specific object. For more information on popup menus see *the Menus and Toolbars* section of this style guide.

Do not use middle mouse button.

Always let the user determine the assigned behavior for the middle mouse button using the mouse utility in the Control Panel.

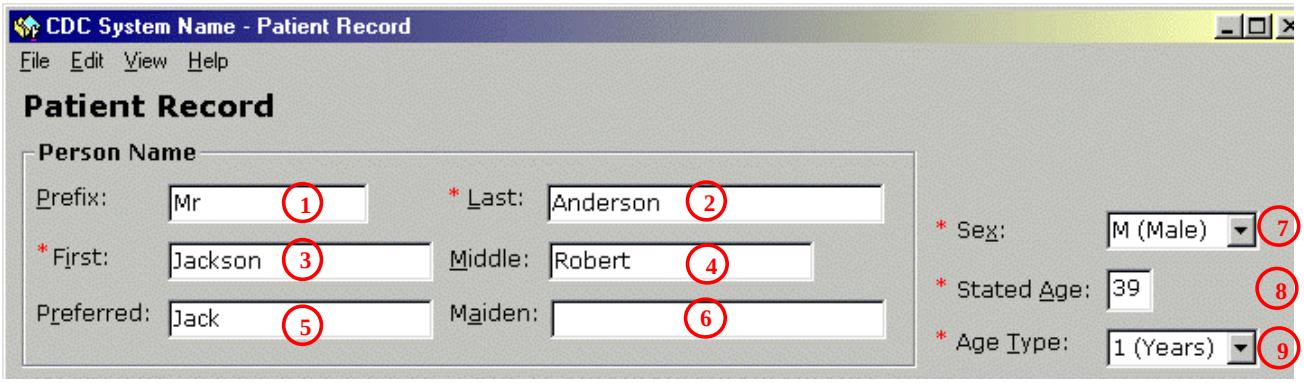
Make single clicking and double clicking consistent.

Use single clicking for selection. Use double clicking for selection plus performance of the default action. For example, double clicking an item in a list box should have the same effect as selecting the item and pressing the enter key.

Establish a tab order from top to bottom and left to right.

The order that controls can be accessed using the tab key has a direct impact on many features of the interface. To insure consistency and correct navigation using access keys, the tab order will be established from the top of the screen to the bottom of the screen starting at the left and moving to the right. Tab orders can be self contained within a grouping as shown in the figure titled *Correct Use of Tab Order*. Sex and Stated Age fields are outside of the Person Name group box and are not considered to be part of the scope of the Person Name tab order.

Correct Use of Tab Order



Make shortcut keys redundant.

Shortcut keys should be an alternative way to access a command - not the only way. Always provide the user with a more visible alternative.

Establish access keys according to CDC standards.

An access key is an alphanumeric key (mnemonic) that is used in combination with the ALT key to navigate and gain access to a control. Access keys will be provided to controls that can handle access keys as was stated in the Controls section of this style guide. When assigning access keys follow these CDC standards:

- ***Make access keys unique within scope***
- ***Assign access keys considering the following (in the order stated):***
  1. first letter of the control label, unless another letter provides a better mnemonic association
  2. distinctive consonant in the label
  3. vowel in the label
- ***Do not assign access keys to lowercase letters with descenders such as g, j, p, q, or y - conflict arises between the access underline and the letter descender***
- ***Do not assign access keys to group boxes***
- ***Do not assign an access key to the OK button since it is selected with the Enter key when it is the default button***
- ***Do not assign an access key to the Cancel button since the Esc key is used to dismiss a modal dialog box***
- ***Follow Window standard access keys for common commands that are documented in The Windows Interface Guidelines for Software Design.***

Use established access keys for common data elements.

Most systems have information that is called by a common name across all areas. When assigning access keys, identify and use those organizational standards already defined for common data elements. It is important that common data be named and accessed in the same way across all CDC surveillance applications.

Establish shortcut keys according to CDC standards.

Shortcut keys, also known as accelerator keys, are key combinations that provide quick access to frequently performed operations. Often times these operations appear in the menus, but can also be used on other frequently performed operations. When assigning shortcut keys follow these CDC standards:

- ***Combine CTRL + a single letter or function key***
- ***Use SHIFT + non-letter key to extend or complement the actions of the key or key combinations used with the SHIFT key. For example, ALT+TAB switches windows in top to bottom order. SHIFT+ALT+TAB switches windows in reverse order***
- ***Use CTRL + key combinations to represent a larger scale effect. For example, HOME moves to the beginning of a line, CTRL+HOME moves to the beginning of the text (document)***
- ***Do not use SHIFT + letter key combinations because the effect of the SHIFT key differs for some international keyboards***
- ***Do not use ALT+ key combinations because they may conflict with access keys.***

The following shortcut techniques are reserved for the common window commands and should not be used:

ALT+ENTER	*displays the property sheet window*
ALT+ESC	*switches between two windows*
ALT+HYPHEN	*displays pop-up menu for active child window (MDI only)*
ALT+SPACEBAR	*displays popup menu for window*
ALT+TAB	*switches window in top to bottom order*
ALT+F4	*closes a window*
ALT+F6	*switch to next window within application*
ALT+PRINT SCREEN	*captures active window image to clipboard*
PRINT SCREEN	*captures desktop image to clipboard*
CTRL+C	*copy*
CTRL+O	*opens primary window*
CTRL+P	*print*
CTRL+S	*save*
CTRL+V	*paste*
CTRL+X	*cut*
CTRL+Z	*undo*
CTRL+F6	*displays next child window (MDI only)*

CTRL+TAB	*displays next tabbed page or child window (MDI only)*
CTRL+ALT+DEL	*reserved to system use only*
ESC	*stops or cancels a function in process*
F1	*displays window with contextual Help information*
SHIFT+F1	*starts context-sensitive Help*
SHIFT+F10	*displays pop-up menu*

Follow CDC navigational terminology standards.

It is very important that the interface prepares the user for what will happen when they initiate the command. The table titled *Navigational Terms* lists the standard command terminology and resulting actions that should occur in all CDC surveillance applications.

Navigational Terms

Name	Action
Logon	Opens a session for the user within the application, and activates all assigned privileges
Logoff	Closes the user session in the application
Apply	Only to be used in Window applications to preview changes made to an option dialog or property sheet.
OK	To confirm a message box, or to apply changes to an option dialog or property sheet. *Not to be used to carry out Add, Edit, or Delete actions.*
Submit	User submits data for processing. Feedback should always follow this command to indicate whether processing was successful.
Add	User sends request to add data to database. Feedback should always follow this command to indicate whether processing was successful.
Edit	User sends request to change data in database. Feedback should always follow this command to indicate whether processing was successful.
Delete	User sends request to delete data from database. Feedback should always follow this command to confirm that this request should take place and then to indicate that it was successful.
Clear	User sends request to clear all fields in form on page.

	page.
Reset	User sends request to restore form on page to previous defaults. This does not clear defaulted fields.
Cancel	User sends request to cancel current action.
Search	User sends request to initiate a search
Close	User sends request to close window, page or file
Exit	User sends request to exit application
Back	Takes the user to the previous dialog window or page in a linear sequence. Should be used by applications in linear dialogs where the user is moving through a sequential dialog.
Next	Takes the user to the next dialog window or page in a linear sequence. Should be used by applications in linear dialogs where the user is moving through a sequential dialogs.

Do not place View commands in command buttons.

Commands that control how the user views the interface should be placed in the View menu. Commands that include "hide", or "show" are best placed in the menu. They can also appear on a toolbar entry if they are frequently used.

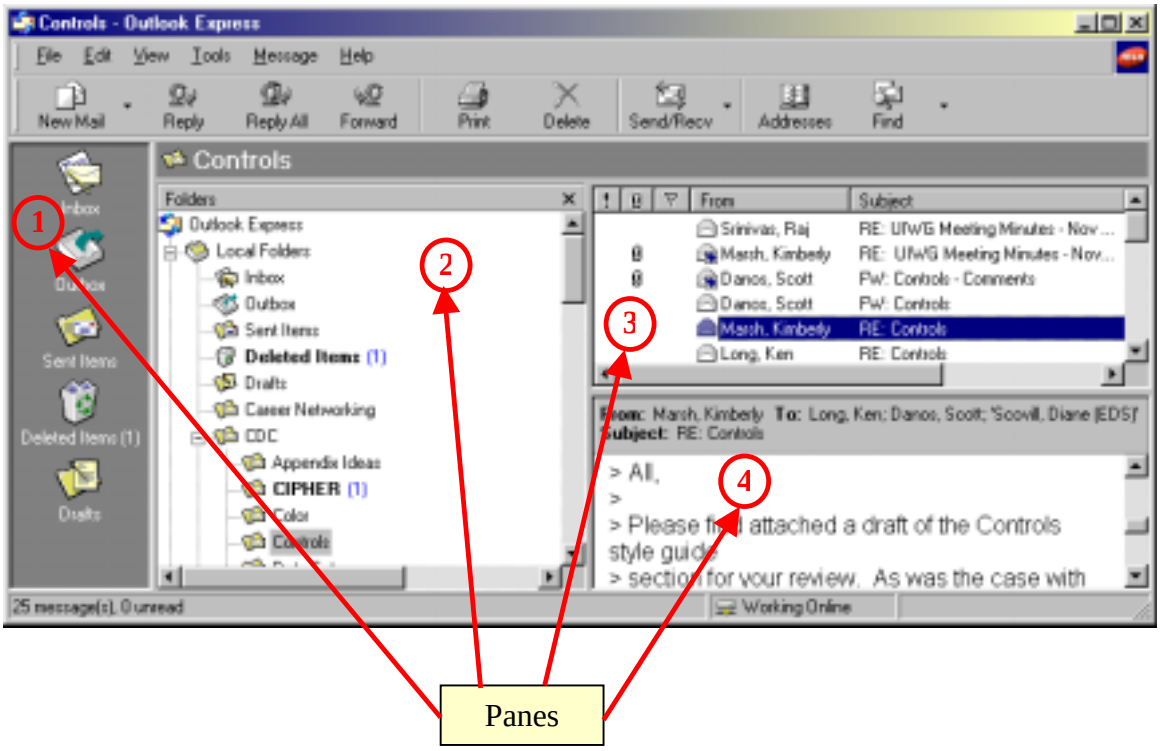
Do not place Exit command in a command button.

Exiting the application will follow Windows guidelines and always be located in the File menu. Do not create command buttons to exit the application. In addition, do not place exit commands in the toolbar. The windows interface provides many exit points through the File menu, the close box, and the application icon pull-down menu. Do not provide any others.

Provide an Apply button on property sheets.

An apply button should always be provided when the user needs to preview the changes that have been made to a property of some object or screen element. Properties or options are usually maintained on what is commonly called a property sheet. The apply button enables the user to experiment with settings. If it the option or property being changed does not have an immediate visual impact to the interface then the OK button can be used.

Panes



Panes allow information to be organized for more efficient viewing. Panes, which can contain a wide variety of controls, including tabs, list bars, and tree views, can enhance the interface design when they are correctly used. However panes can also greatly confuse the user and should be used only after considering the skill set possessed by the user, as well as the method by which the information is organized.

The Outlook example above shows just one of the many possibilities involved in creating panes. In this example there are 4 panes of information each using a different control. Developers should pick the appropriate controls only after they have a thorough understanding of how the information is related to the user task.

Panes will be organized in a top to bottom, left to right progression.

Each pane in the design is meant to enhance the next pane. This is done by organizing the navigation in a hierarchical way. For example, when a new view is chosen in pane 1 (Outlook bar), it affects panes 2, 3, and 4. When a new node is chosen in pane 2 (Folder tree), it impacts the panes 3 and 4, but not the pane 1. When an item is selected in pane 3 (mail items list view), it only impacts the pane 4. This is done to bring order to the set of panes and follows rules of progressive disclosure that are part of user centered design.

Panes that are organized independently and do not follow this progressive navigation scheme will only confuse the user. Developers who embrace multi-pane applications need to be very careful to ensure that the information is well organized and that the panes have clear relationships.

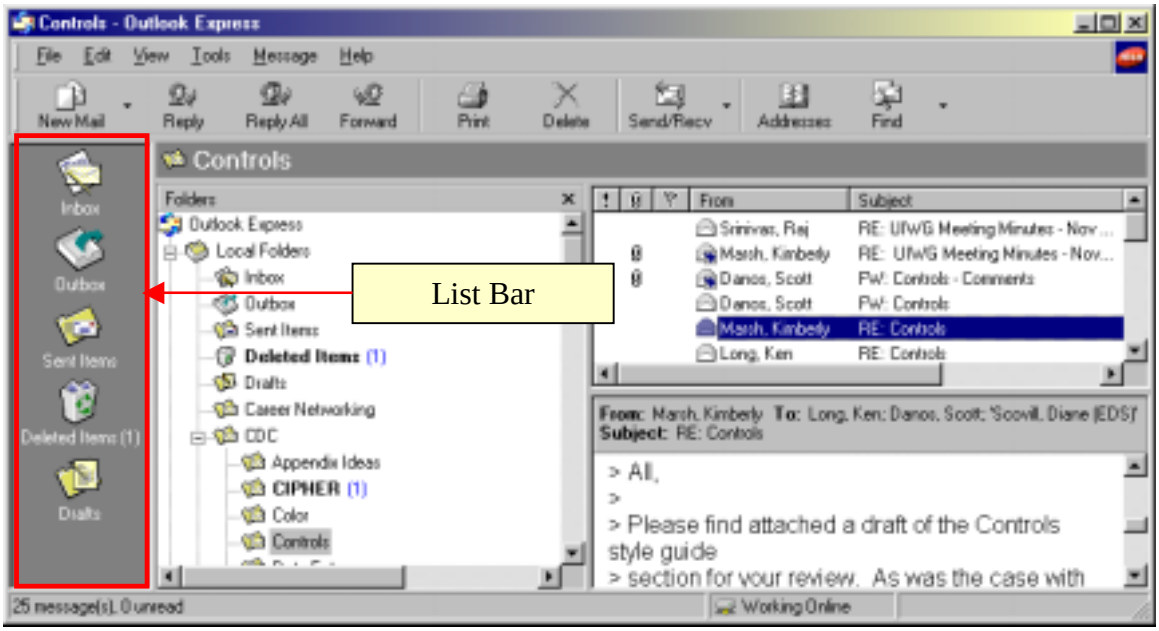
Use no more than 4 panes.

The greater the number of panes, the more challenges the user will have in processing the information as presented on the screen. For CDC applications, no more than 4 panes should be used. Five or more panes is likely to lead to confusion among users and therefore should be avoided.

Panes should be user configurable.

When panes are used as an alternative to navigation that is present in the menu or toolbar, the user should have the ability to remove or add those panes to the interface.

List Bar



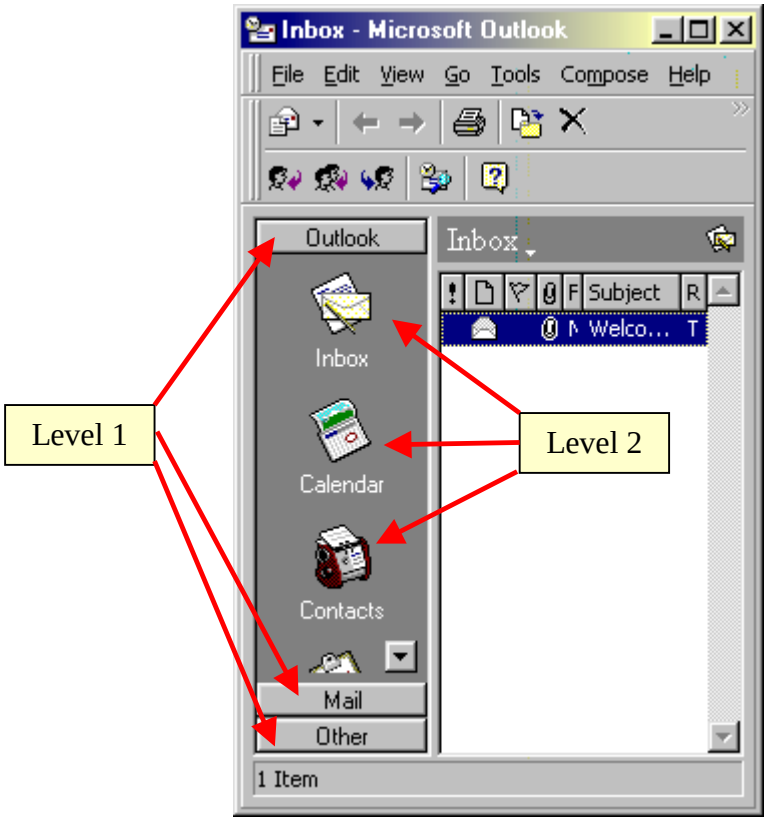
The list bar, most commonly seen in the Windows interface as an Outlook Bar, is an alternative navigation tool for the user. The list bar is changing the standard Windows navigation model to align it more closely with the Web browser navigation model.

List Bars, like the Outlook Bar shown in the figure above, list the important places where a user is likely to want to go. These bars tend to be very customizable and represent high level categories of navigation that are useful for the user. While list bars are becoming more popular, they have their drawbacks when used with too many panes, sliding groups, or command buttons.

Use listbars to organize navigation shortcuts in 2 levels..

The purpose of the list bar is to supply the user with navigation shortcuts. It's actually a shortcut bar. As illustrated in the figure titled *List Bar Navigation Levels*, the information represented in the sliding group button is the first level of navigational grouping, the text and icons represent the second level shortcuts to information shown elsewhere in the interface. Include only those navigational choices that are most often used and allow the user to customize based on their perceptual model.

List Bar Navigation Levels



Make list bars user configurable.

All list bars should be able to be customized by the user by using the standard drag and drop navigation that is evident in the Outlook application. Users approach their tasks differently and need to be able to customize their shortcuts to assist in expediting their tasks.

Make all list bars vertically oriented.

All CDC surveillance applications will orient the list bar down the window vertically.

Use solid windows standard dark grey as the background color.

All CDC surveillance applications will use the windows standard dark gray as the background color for all list bars.

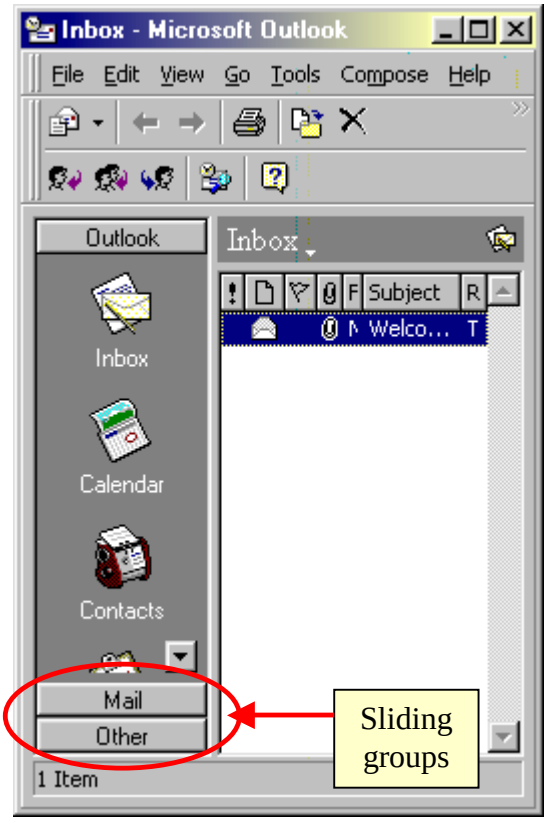
Avoid shortcuts to external applications.

List Bars allow you to define shortcuts that can launch external applications such as email. For CDC applications avoid launching these external applications unless their use is directly tied to the object being viewed. For example, if you are viewing patient information and you want to launch an email directly to the patient being viewed by automatically populating the email header with the patients email and subject information, then launching an external email application makes sense. But, do not launch an email application simply as a shortcut.

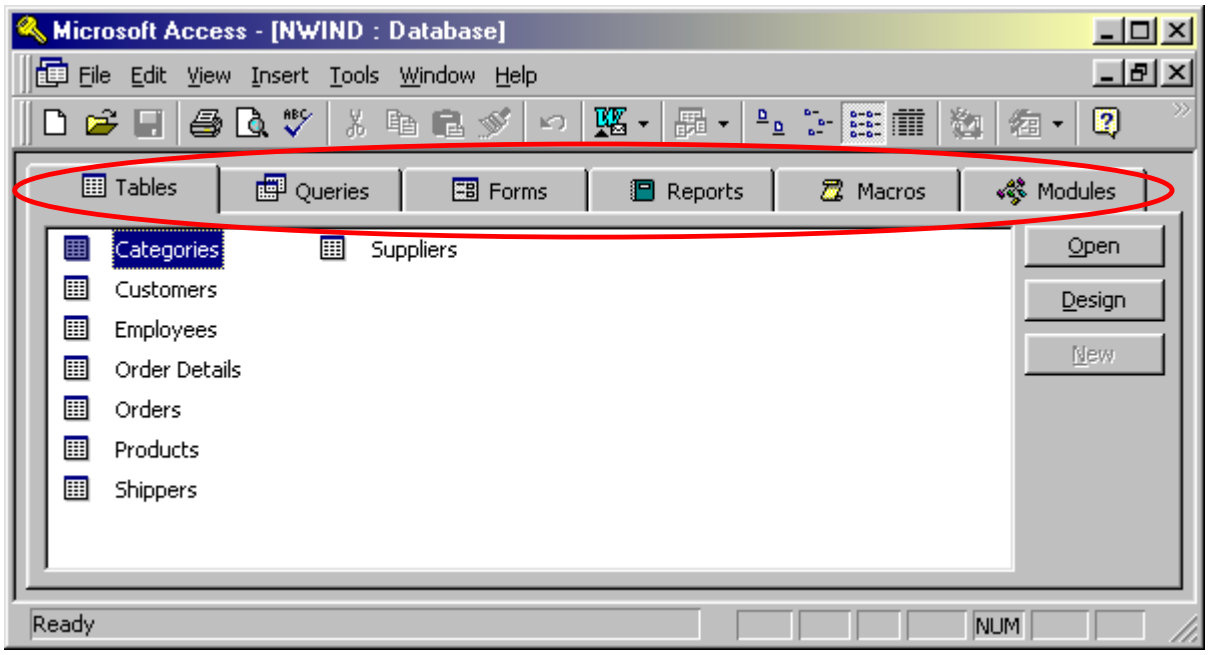
Minimize sliding groups.

Sliding groups are achieved through the use of the sliding group buttons as shown in the figure titled *Sliding Groups*. These allow partitioning of the navigation shortcuts. When pressed the sliding group button shows a different group of shortcuts within the list bar. Do not create categories of shortcuts from external applications. Take care in creating and naming the sliding groups. Strive for no more than 7 sliding groups. This supports short-term memory constraints and the need for progressive disclosure.

Sliding Groups



Tabs



Tabs proliferate the Windows interfaces of today. Tabs are an easily understood metaphor and should be used when the user's tasks require that information be put into categories and be viewed in only 2 levels. In addition, tabs should also be used only when information does not have to be accessed in any particular order. An example of appropriate tab use in a main application can be seen in the Microsoft Access screen above. Tabs are commonly used on option windows, called property sheets, as well to break down the multitude of options that are available to the user into categories.

Use 2-level tabs only.

Users should never click on information in a tab that results in the appearance of yet another group of tabs. Tabs should only be used when the user's tasks require that information be put into categories and be viewed in only 2 levels.

Use tabs to group information rather than to display grouped functions and commands.

Tabs should be used to present groups or categories of information to the user such as patient demographics, symptoms, or question categories from a form. Tabs should not be used to group functions or commands. Functions and command groupings are best shown in menus.

Use tabs for grouping nonsequenced information.

Tabs should be able to be accessed in any order. When information must be presented to the user in a specific order for them to complete their task, tabs are a poor choice. Even if the tabs are numbered, there is nothing to prevent the user from selecting a tab out of

order and there shouldn't be anything that restricts the user to a particular order when it comes to tabs. Tabs can always be used to access information in any sequence.

Use tabs to group static categories of information.

When the categories of information remains fixed, use tabs. If these categories are dynamic and can change based on user action, avoid tabs. Tabs that shrink and grow with ever changing rows only confuse the user. In addition, tabs that can grow to an undetermined amount may end up creating an unmanageable amount of rows. Keep tabs fixed whenever possible.

Display tabs across the top of the window or pane.

To ensure consistency across CDC surveillance applications, all tabs will be shown at the top of the interface as opposed to the bottom or side.

Do not display a single tab.

The purpose of tabs is to categorize information in a single view. If there are not multiple categories of information present, do not show a tab as is illustrated in the figure below. The title bar provides the same information that is in the tab. In this case the single tab is redundant and unnecessary.

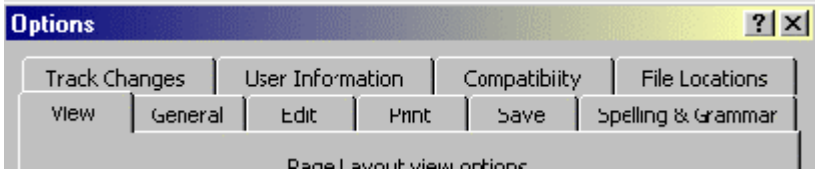
Do not use more than two rows of tabs.

A single row of tabs is preferred, but two rows are acceptable. On a main application window, stick to 1 row of tabs if possible. Property sheets or options dialogs are where 2 rows of tabs come in handy. More than two rows of tabs is too complex and makes navigation confusing. Use multiple dialog boxes instead.

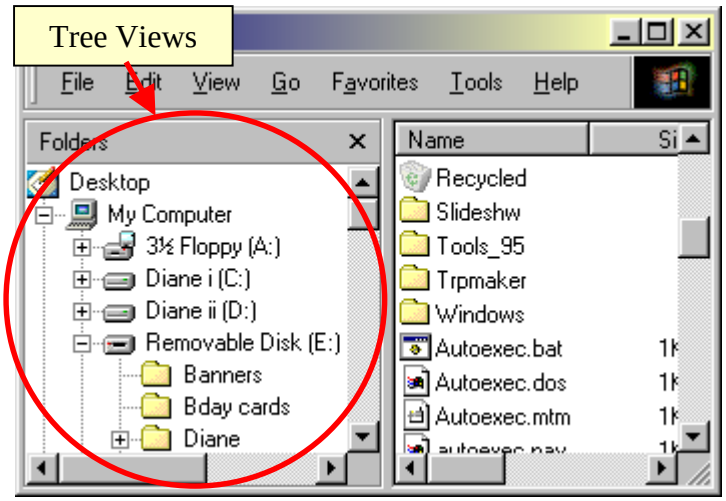
Size tabs to fill the width of the window or pane.

Tabs should be sized so that their row fills the width of the window or pane. Notice in the figure titled *Tab Width Example* that the tabs in each row have been sized so that the rows fill the dialog. This provides a smoother and more legible interface.

Tab Width Example



Tree Views



Tree controls display a set of objects in an outline format that is based on a hierarchy. The hierarchy implies that there is a relationship between the objects. Trees are most appropriate for tasks in which the user is required to think about information and how it relates to other information. While each element of the tree can be thought as a grouping of data, the key point is that the tree should be used when it is important to show the relationship between the groups. The tree is also best when it is necessary to see the relationships in a single view.

Use trees for hierarchical presentation of information.

When there are several groups of data that a user must access, and each of those groups has 1 or more subgroupings, use a tree. In addition, use the tree when it is important for the user to see all of the relationships in a single view.

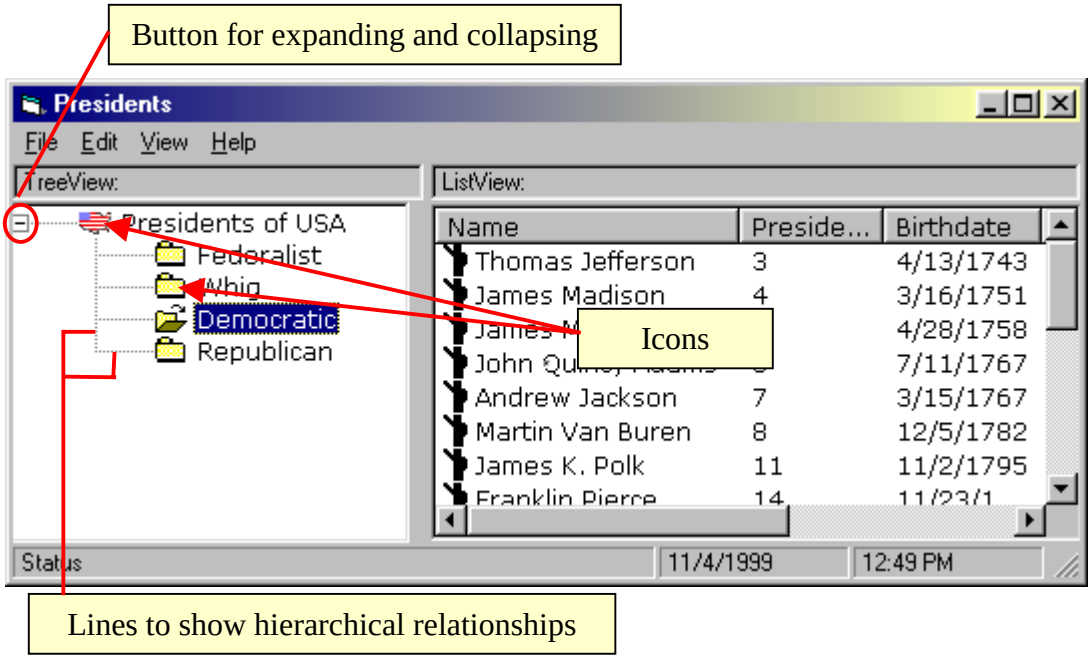
Label all tree views.

To provide more understanding of the interface panes and their organization, all tree panes will be labeled.

Include icons, buttons and lines in all tree view displays.

The tree view control is particularly useful because it has buttons that allow the user to expand or collapse an element. It provides the user the ability to directly manipulate the view of any of the elements that are displayed. Although icons, buttons and lines are an optional behavior in this control, it is best to include these features because they make it easier for the user to interpret the hierarchy. These features are illustrated in the figure titled *Treeview Example*.

Treeview Example



General Navigation and Metaphor Guidelines

The following guidelines have been established within the user interface design community and should be applied across CDC software and web-based surveillance applications using HTML.

Make navigation easily learned.

If the user spends too much time having to learn a complex navigation, they won't have much energy left to get their task done. In navigation design, it's best to avoid burdening the user with a steep learning curve. Excessive learning curves have a negative impact on the user's satisfaction.

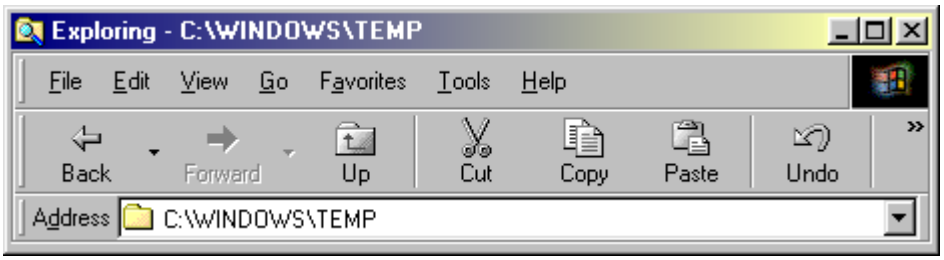
Make navigation consistent.

Make sure your approach to navigation is consistent or you will confuse your users. The ability to predict where navigational elements will be found is an important first step in understanding software or a web site. People will put up with a few navigation quirks as long as you are consistent in your offerings, their placement, and their appearance.

Make navigation appear in context.

Whether in a software package or on the web, make it clear to the user where they are, whey they have been, and how to get back to where they came from. Showing context prevents users from getting lost. For example, Windows Explorer and Internet Explorer use this technique in giving the user the option to display an address bar that shows what folder or URL they are currently viewing, as shown in the figure titled *Windows Explorer Showing Context*.

Windows Explorer Showing Context



Avoid complexity.

Software or web sites that feature layer upon layer of subcategories with many levels to click through will induce the "Are We There Yet?" syndrome. Other structures that contribute to this are:

- *Ridiculous number of steps to complete before any serious content can begin.*
- *Forms that span several screens or pages*

Navigation should provide feedback.

Feedback is often the only way that users can tell whether they have been successful. Pick controls that are responsive and provide information about location. Both of these types of feedback are essential in helping users move around software or web sites.

Don't bend the interface to fit some arbitrary metaphoric standard.

The heavy reliance on metaphor means that you can intuit the basic functioning of the software, but the downside is that the metaphor can restrict navigation to a very rudimentary, linear path. The software can end up relying exclusively on metaphor for every aspect of its interface. In the world of software there is no reason to constrain the user with real-world limitations. Abandon the devotion to metaphor and provide the user with services they can't get out in the real world. Be cautious when stretching the metaphor beyond its real-world limitations, it may confuse the user.

Make shortcut keys user configurable when possible.

Provide support for allowing the user to change the shortcut key assignments in your software, when possible.

HTML Specific Navigation and Metaphor Standards

The following standards are to be adopted across all CDC web-based applications using HTML. These standards are intended to lead to a more consistent and usable interface.

Always provide a home page.

The user can easily get lost on the web. Provide a home page for your web site or application with a URL that is easy to remember. When a CDC application is released to the user, the documentation must provide the URL and encourage the user to bookmark the home page so they have an easy way to return should they get lost.

Always specify the CDC surveillance web application name in the <title> tag

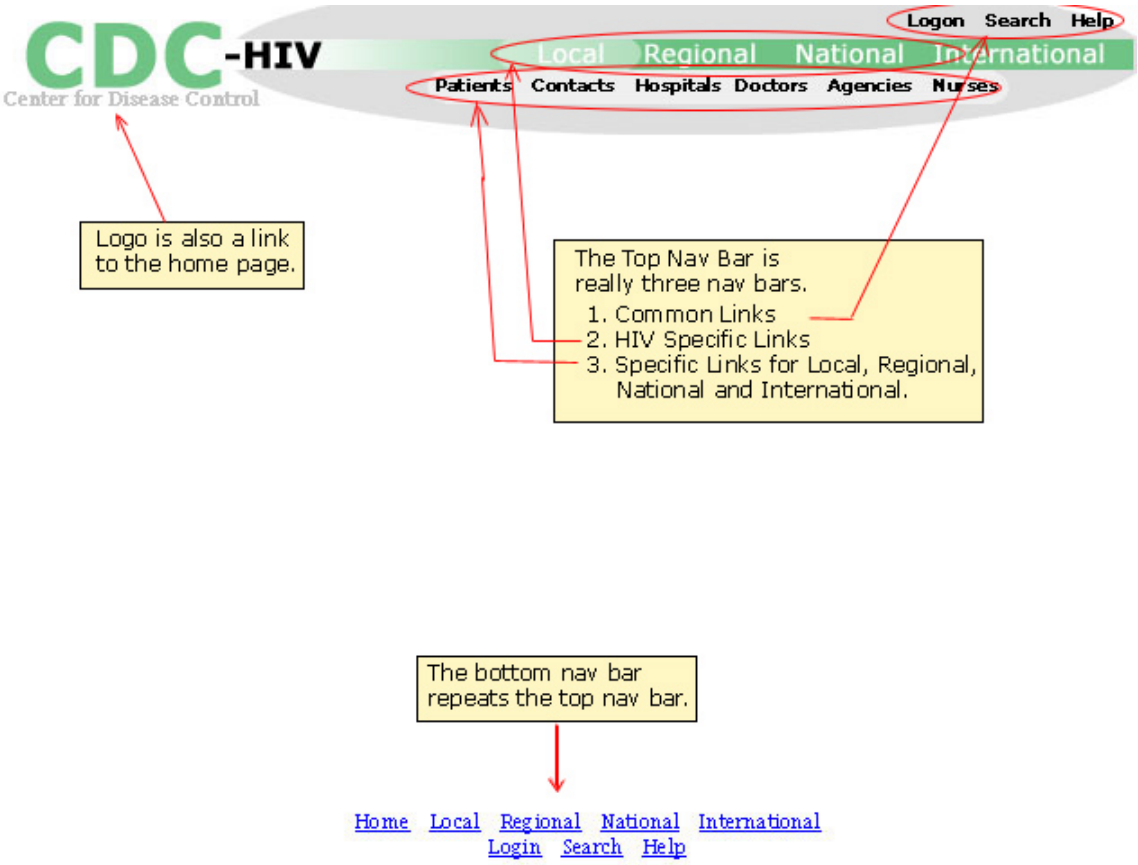
The text in <title> tag is displayed in the title bar of the browser window. The text and the URL are saved when the user bookmarks the web page.

Use CDC survelleillance application logo in upper left corner and link to home page.

Users often bookmark pages in web sites. The link to the home page is extremely valuable to the user later when they return to that page, and want to access information that was on another page of the web site. The logo will be in the top left corner of the web page and will be a link to the home page. CDC system logos are being developed for Windows splash screens. A comparable one should be available for use on web pages.

Refer to the figure titled *Nav Bars Example* for standards involving nav bars.

Nav Bars Example



Provide navigational bars at the top of every CDC surveillance web application.

Refer to the figure titled *Nav Bars Example* above. There are three nav bars at the top of the web page. The first nav bar is a short nav bar with links that are common for all CDC surveillance applications. This nav bar will contain no more than 5 entries. The second nav bar is specific to the application (analogous to a menu bar in Windows) and will contain no more than 9 entries. The third nav bar appears when the user selects a link from the application-specific nav bar (analogous to entries on a drop-down list on a Windows menu bar) and will contain no more than 9 entries.

Provide navigational bars at the bottom of every CDC surveillance web application if the user must scroll down a page.

Refer to the figure titled *Nav Bars Example* above. The bottom nav bar contains text links mirroring the links on the top nav bar so the user does not have to scroll to the top of the page to navigate.

Avoid use of frames in web-based surveillance applications

Frames have drawbacks for the user. The browser back button and bookmarks link to the beginning of the frameset, not the frame where the user may expect. Frames use more screen real estate and take longer to load. Frames can be tricky for developers to use correctly. CDC surveillance application programs will not use frames. If a CDC Windows surveillance application using panes is replicated to the web, it should use window-like controls rather than frames**.**

Do not use command buttons in place of links or vice-versa

Clicking on a command button initiates a server-side action. Clicking on a link takes the user to another page. Do not visually confuse the user with inappropriate use of command buttons and links.

Provide links to logon and logoff pages for CDC surveillance web-based applications that have restricted user access

Web users typically enter and leave web sites without logging in or logging off. Web-based applications may require user id and password access in order to grant the user appropriate privileges within the application. Where this is the case, it is necessary to require the user to logon and logoff. "Logon" and "Logoff" should be links to "logon" and "logoff" pages, respectively, where the user can click on the appropriate command button.

HTML Specific Navigation and Metaphor Guidelines

For web based applications using HTML, the following guidelines are recommended and should be applied across CDC web-based applications using HTML.

Create a broad, shallow navigational scheme

Do not force the user to drill down through many intermediate pages to reach their objective. A broad, shallow structure is preferred over a narrow, deep structure. Strive to have the user reach their objective in 3 clicks or less.

Provide several different navigational elements

Different people have different approaches to find things, so provide multiple ways for users to navigate through your web site. The following list describes some common navigational devices. *Those marked with (*) are generally expected on all web sites*. Many web sites provide a combination of navigational devices all on the same page.

Organization (*)
Although it is not a navigational element, the organization of the web site contributes greatly to its ease-of-use so it should be well planned and usability tested before navigational schemes are developed. A well-organized web site makes it easy for the user to form a mental model of the web site allowing them to intuitively find what they want.

Nav bars (*)
Navigational bars (commonly referred to as "nav" bars) have become a de facto standard on the Web and are a standard for CDC surveillance web-based applications. A nav bar may be a row of text links, a row of graphics, or a row of graphics and text. The names of the links should reflect the organizational structure of the web site. The number of links in a nav bar is limited by the available screen real estate, and the size of the link. It is especially important that the nav bar is easily readable. Many nav bars have so many entries and are presented in such a small font that they are very difficult to read. Nav bars are preferred when screen real estate is an issue. Nav bars should never require the user to scroll to the right. Top nav bars are generally graphical. Bottom nav bars are generally textual.

Side bars
Side bars are columns of links along the left or right side of the page, visually separated from the rest of the page. The links reflect the organizational structure of the web site. If a CDC surveillance application uses a side bar it should be only to supplement nav bars.

Navigational paths
Some web pages provide a navigational path showing the series of pages that led the user to the current page. Each page is generally separated by a colon or arrow and is a link back to that page. The path is generally at the top of the page above or below the nav bar.

Site(*)
A site map provides a high-level view of the web site, reflecting the organizational structure of the site. It is very similar to the table of contents of a book, where each item in the table of contents is a link to the appropriate page.

Index
An index is very similar to an index in a book, providing an alphabetized list of words and topics in the web site, where each word or topic is a link

to the appropriate page. Very few web sites provide an index, even though it is very useful for the user. An index is time consuming to create and even more time consuming to maintain.

Search
Many web sites provide a search where the user specifies keywords to search for, then clicks on a command button to start the search. The results of the search are displayed and each result includes a link to the appropriate page within the web site. The search may be limited to portions of the web site or have filters narrowing the results.

Menus
Menus on the Web are not the same as the Windows menu bar. In the early days of the Web, menus consisting of a list of links presented in the center of the screen were the only form of navigation. Today, menus are still used very frequently, often clustered together in boxes, to direct the user to specific portions of the web site**.**

Links (*)
When a user clicks on a link, text or graphical, they expect to be transported somewhere else. The "somewhere else" may be on the web site or somewhere totally different.

Help the user by providing information that will give them a clue as to where they may go when they click on the link. The information may be a textual phrase or a little icon. Some web sites use little globe icon after a link to indicate that the user will leave the web site if they click on the link. Some web sites use a little book icon after a link to indicate that the user will go to a glossary on the web site**.**

Command buttons
Command button are very different from a link. When a user clicks on a command button they expect that some server-side action will take place. The command button contains a label indicating the server-side action that will take place. It is very important that the interface prepares the user for what will happen when they click on the command button, and where they will end up when the server-side action is completed. The following table lists the standard command buttons and resulting actions in all CDC surveillance systems.

Frames
Frames on the Web are look similar to panes in Windows. Each pane (or frame) is really a web page. Several frames combine to appear as a single web page to the user. Content in one or more frames usually remains fixed, while the content in other frames changes based on what the user selects. Frames unfortunately have a number of drawbacks. Frames occupy more screen real estate that a typical web page and takes longer to download. When the user clicks on the browser back button, he is returned to the beginning of the frameset, not the previous page. When the user sets

a bookmark for a framed page, the bookmark really links to the frameset, the initial web page displayed. The user may have linked deep into the frames, but the bookmark will return him to the first page. Frames can be tricky for developers to control, resulting in pages that look like frames within frames.

Recommended Readings

McKay, Everett N. *Developing User Interfaces for Microsoft Windows*. Microsoft Press, 1999
Chapter 4 - Establishing a Consistent User Interface Style
Chapter 10 - Good Interfaces are Invisible
Chapter 29 - Check Your Dialog Boxes

Fleming, Jennifer. *Web Navigation: Designing the User Experience*, O'Reilly & Associates, Inc., 1998

LeMay, Laura. *Graphics and Web Page Design*, Sams.net Publishing, 1996

Morville & Rosenfeld. *Information Architecture for the World Wide Web*, O'Reilly & Associates, Inc., 1998

The Windows Interface Guidelines for Software Design. Microsoft Press, 1995
Chapter 4 - Input Basics
Chapter 5 - General Interaction Techniques
Chapter 8 - Secondary Windows
Appendix A - Mouse Interface Summary
Appendix B - Keyboard Interface Summary

Helpful Web Sites

Kirsanov, Dimitri, *Designing Site Navigation*,
http://www.webreference.com/dlab/9705/

Marcus, Aaron, *Managing Metaphors for Advanced User Interfaces*. ACM Digital Library, http://www.acm.org

MacWorld article by Peter Norville, *Worst Web Faux Pas*
http://macworld.zdnet.com/1999/02/create/web.html

Neilsen, Jakob, Alert Box article *Features for the Next Generation of Web Browsers*
http://www.useit.com/alertbox/9507.html

The Webreview.com *Navigation and Usability Guide*.
http://webreview.com/wr/pub/98/05/15/thing/index.html

Smilowitz, Elissa D., *Do Metaphors Make Web Browsers Easier to Use?,* ACM Digital Library , http://www.acm.org